ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

A New Approach for Encryption Method using Collective Techniques with Rijndael Algorithm

S.Devi[1], K.Kanagaram[2], V.Palanisamy[3]

Research Scholar, Computer Science & Engineering, Alagappa University, Karaikudi, India[1]

Research Scholar, Computer Science & Engineering, Alagappa University, Karaikudi, India[2]

Professor & Head, Computer Science & Engineering, Alagappa University, Karaikudi, India[3]

Abstract: Network security is the best known mechanism in the computer world, which secures the data against hackers or unauthorized users over the communication channel. In this paper three mixed techniques are used to enhance the security i.e. One is Rijndael, other is cross change & transformation followed by basic arithmetic and logic operations. In the existing system, the encryption is less secure & less complex while the proposed work implements the size of the private key for ensuring the data protection. Data secrecy is mainly dependent on the strength of key size in the encryption algorithm. The proposed algorithm computes a random key sequence for each & every plaintext character. So that the encrypted data should be in the disguised format to analyze.

Keywords: Rijndael algorithm, random key sequence format, cross change, transformation, encryption, decryption

I. INTRODUCTION

Network security measures are needed to protect data during the transmission. The term network security is somewhat misleading, because virtually all business, government and academic organizations interconnect their data processing equipment with a collection of interconnected networks. In developing a particular attack on those security mechanism or algorithm, potential attacks must be considered on those security features. In many cases, successful attacks are designed by looking at the problem in a completely different way, exploiting an unexpected weakness in the mechanism [1].

Cryptography is the art of secret writing [2]. There are two ways of changing message in cryptography to confuse anyone who intercepts it: codes and ciphers. A code replaces each word with another word that has a different meaning. Ciphers convert the message by a rule, known only to the sender and receiver [3].

With the help of cryptography, it is possible to communication securely through insecure channels. It has two types in nature, one is symmetric key cryptography and another is asymmetric key cryptography. The first one uses the same key for both encryption and decryption process while the second uses different keys for encryption and decryption process. A symmetric key model is referred as private key cryptography. Asymmetric key model is defined as public key cryptography [3].

A. Substitution & Transposition

Substitution and Transposition methods are widely used cryptographic techniques in the network security. Substitution method involves replacement of a character by another one whereas in transposition the positions of characters are changed accordingly [4].

B. Genetic Algorithm

In this paper genetic programming concepts are used to increase the security risks of messages. Generally, the crossover and transformation techniques are applied to the cryptographic algorithms for alteration of plaintext characters position. Genetic algorithms are adaptive heuristic search algorithms based on mechanisms of natural selection and natural genetics. Cross change or Crossover generates new one or more individuals (children or offspring) from the two given points (parents). Transformation is the mutation operator which randomly changes the characters in the individual (resulted from the crossover technique) [5].

II. RELATED ALGORITHMS

Cryptography is the one the gives a right way to secure the data while in transmission. The task needed to attain such security is referred as encipherment. Today's computer world, various encryption algorithms are available. Most of them concentrate mainly on the encryption key. To avoid the cryptographic attacks, an encryption key is taken as much length (bits) as possible. In the cryptographic history, whether the key is chosen from the plaintext character or create an automatic/random key generation algorithm.

An encryption algorithm presented by S.G.Srikantasamy and H.D.Phaneedra uses the concepts of arithmetic & logic operations. The key is determined from the message itself [6].

The introduction of one time pad cipher in the encryption algorithm prevents the hackers over the communication channel. Because the same pad value should be added with the message in both encryption and decryption process. Whenever an unauthorised user needs to compromise the system, they must know a unique pad value. Arithmetic & logic operations are combined with one time pad cipher in the year 2013 by the same author [7].

Traditional Caesar cipher has taken only 26 alphabetical letters for plaintext message and key value. On this basis, attackers could retrieve the encrypted message in easy way. There are only 26 possibilities for encrypting the plaintext message. To overcome this difficulty a modified

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

approach to the shift cipher is made by S.G.Srikantasamy and H.D.Phaneedra in the year 2012 [8].

The second method involved in the key selection technique is introduced by B.Bazith Mohammed. The key is generated automatically and combined with the Caesar cipher method to enhance the substitution technique [9].

In nature, the combinations of techniques are undertaken for scrambling the data into unknown format. Govind Prasad and his team had focused their view on the combined effect of substitution and transposition techniques in encryption algorithm [10].

Recently, Devendra Prasad [11] has presented an encryption algorithm which extends the key size up to maximum ASCII values.

In that paper, basic arithmetic and logic operations are associated with that automatic key generation technique. But in the security aspect, a constant key is not enough to encrypt or decrypt a message.

III. PROPOSED WORK

The proposed algorithm takes random number sequence for each end every character. Random key value is different for every message. The proposed work implements the security level through various stages. In this paper, the algorithm is categorized into three stages. On each stage an intermediate cipher text is produced at the process end. In this encryption algorithm two intermediate ciphers are produced at first process, final cipher text is resulted at the third process.

A. Flowchart for Encryption

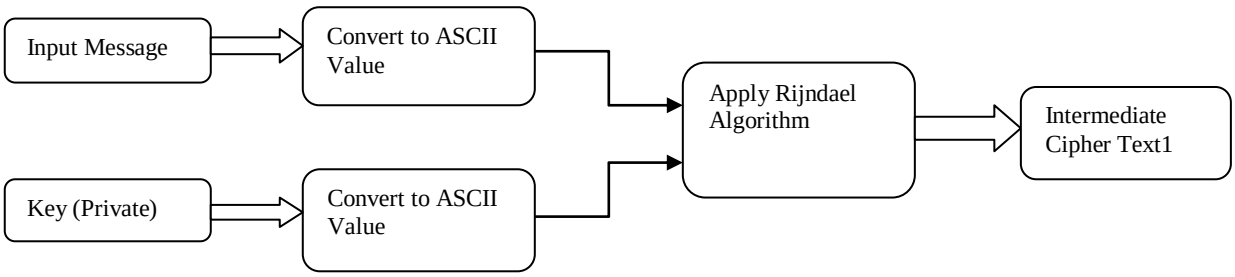

Fig 1: Process 1

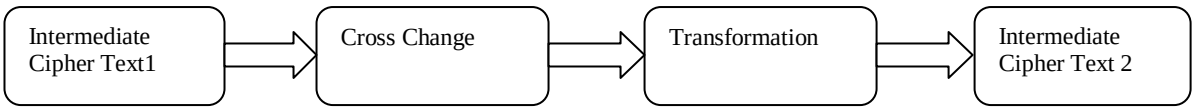

Fig 2: Process 2

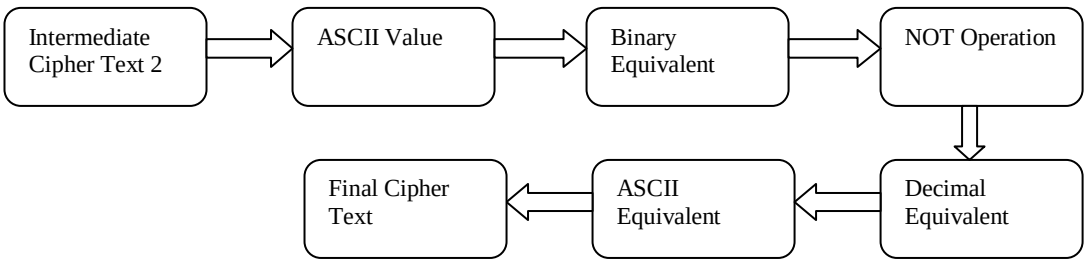

Fig 3: Process 3

B. Flowchart for Decryption

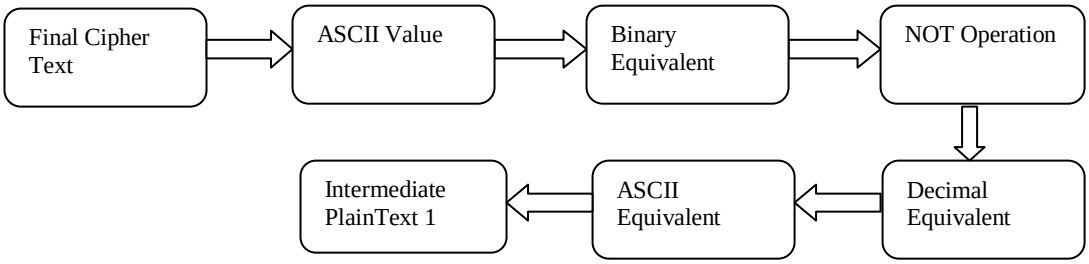

Fig 1: Process 1

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

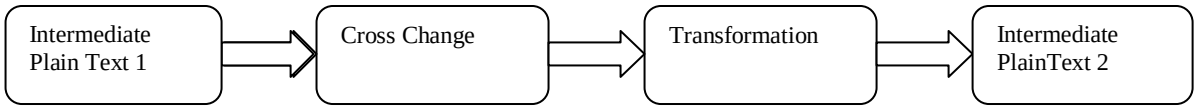

Fig 2: Process 2

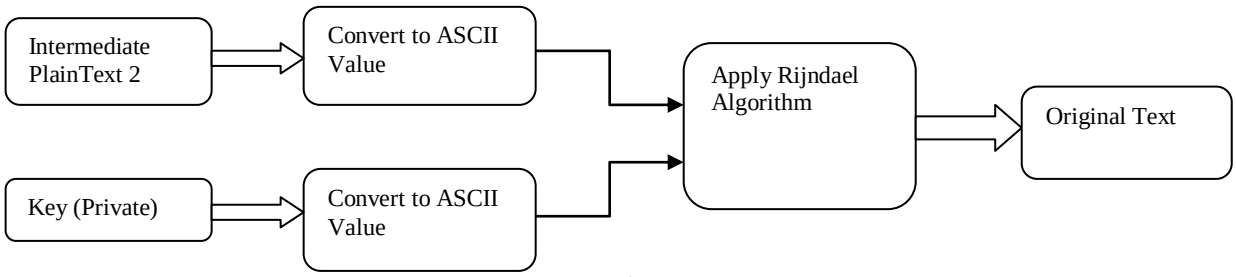

Fig 3: Process 3

C. Encryption Algorithm

1) Process:

- Consider the plaintext message as the input.
- Generate a random key value as the message length.
- Convert each character in the input and key value into its corresponding ASCII value.
- Apply Rijndael algorithm to them.
- At the end of process I, an intermediate cipher text 1 is produced.

2) Process:

- Take the intermediate cipher text 1 as the input.
- Apply genetic programming techniques like cross change and transformation.
- At the end of process II, an intermediate cipher text 2 is produced.

3) Process:

- Take the intermediate cipher text 2as the input.
- Convert each character in the intermediate cipher text 2 into its equivalent ASCII value.
- Converted ASCII equivalent is transformed into its corresponding binary value.
- To shift the positions, apply NOT operation.
- Convert the result into its decimal equivalent.
- Corresponding ASCII equivalent is taken as the final cipher text.

D. Decryption Algorithm

1) Process:

- Take the Final cipher text as the input.
- Convert each character in the Final cipher text into its equivalent ASCII value.
- Convert the result into its decimal equivalent.
- Take the NOT operation to the decimal equivalent.
- Convert the resultant value into its corresponding binary value.
- Convert the binary value into its equivalent ASCII value.
- Take the corresponding ASCII character as the intermediate plain text 1.

2) Process:

- Take the intermediate plain text 1 as the input.
- Apply genetic programming techniques like cross change and transformation.
- At the end of process II, an intermediate plain text 2 is produced.

3) Process:

- Take the intermediate plain text 2 as the input.
- Take the (private) random key value as used in the encryption process.
- Convert each character in the input and key value into its corresponding ASCII value.
- Apply Rijndael algorithm to them.
- At the end of process III, original message is retrieved.

E. Results and Discussion

The input of the encryption algorithm takes “Transmit” as the original message and the private key for that process is “83614957”.The proposed algorithm has long bits of encrypted data even for the small input message. For example8 bits character input message gives minimum of 24 bits character encrypted data. Rijndael of AES algorithm is involved to substitute each plain text into another character by using salt values. The salt values changes or mix up with the plain text characters to produces lengthy altered characters as the intermediate plain text 1.

Example for Encryption algorithm:

TABLE 1

PROCESS 1 USING RIJNDAEL ALGORITHM

Sl. No	Input Message	ASCII Value	Key	ASCII Value
1.	T	84	8	56
2.	r	114	3	51
3.	a	97	6	54
4.	n	110	1	49
5.	s	115	4	52
6.	m	109	9	57
7.	i	105	5	53
8.	t	116	7	55

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

TABLE 2

INTERMEDIATE CIPHER TEXT 1

Result of Rijndael Algorithm
5
L
Y
H
u
w
x
m
y
A
A
P
Q
J
f
+
9
1
D
K
P
w
=
=

At the end of Process 1, an intermediate cipher text 1 is 5LYHuwxmyAAPQJf+91DKPw==

TABLE 3

PROCESS 2 USING CROSS CHANGE & TRANSFORMATION

Sl. No	**Intermediate Cipher Text 1**	**Cross Change by even position**	**Cross Change by odd position**	**Transformation 1**	**Transformation 2**
1.	5	L	5	5	Q
2.	L	H	Y	H	+
3.	Y	w	u	Y	f
4.	H	m	x	L	J
5.	u	A	y	u	9
6.	w	P	A	m	K
7.	x	J	Q	x	D
8.	m	+	f	w	1
9.	y	1	9	y	P
10.	A	K	D	P	=
11.	A	w	P	A	=
12.	P	=	=	A	w
13.	Q				
14.	J				
15.	f				
16.	+				
17.	9				
18.	1				
19.	D				
20.	K				
21.	P				
22.	w				
23.	=				
24.	=				

At the end of Process 2, an intermediate cipher text 2 is 5HYLumxwyPAAQ+fJ9KD1P==w

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

TABLE 4.

PROCESS 3 USING BASIC ARITHMETIC & LOGIC OPERATIONS

Sl. No	Intermediate Cipher Text 2	ASCII Value	Binary Equivalent	NOT Operation	Decimal Equivalent	ASCII Equivalent
1.	5	53	00110101	11001010	202	Ê
2.	H	72	01001000	10110111	183	.
3.	Y	89	01011001	10100110	166	¦
4.	L	76	01001100	10110011	179	³
5.	u	117	01110101	10001010	138	Š
6.	m	109	01101101	10010010	146	’
7.	x	120	01111000	10000111	135	‡
8.	w	119	01110111	10001000	136	ˆ
9.	y	121	01111001	10000110	134	†
10.	P	80	01010000	10101111	175	¯
11.	A	65	01000001	10111110	190	¾
12.	A	65	01000001	10111110	190	¾
13.	Q	81	01010001	10101110	174	®
14.	+	43	00101011	11010100	212	Ô
15.	f	102	01100110	10011001	153	™
16.	J	74	01001010	10110101	181	µ
17.	9	57	00111001	11000110	198	Æ
18.	K	75	01001011	10110100	180	´
19.	D	68	01000100	10111011	187	»
20.	1	108	01101100	10010011	147	“
21.	P	80	01010000	10101111	175	¯
22.	=	61	00111101	11000010	194	Â
23.	=	61	00111101	11000010	194	Â
24.	w	119	01110111	10001000	136	ˆ

At the end of Process 3, the final cipher text is Ê.|³Š,‡ˆ†-¾¾®Ô™µÆ´»“¯ÂÂˆ

Example for Decryption:

TABLE 5

PROCESS 1 USING BASIC ARITHMETIC & LOGIC OPERATIONS

Sl. No	Cipher Text	ASCII Value	Binary Equivalent	NOT Operation	Decimal Equivalent	ASCII Equivalent
1.	Ê	202	11001010	00110101	53	5
2.	.	183	10110111	01001000	72	H
3.	¦	166	10100110	01011001	89	Y
4.	³	179	10110011	01001100	76	L
5.	Š	138	10001010	01110101	117	u
6.	’	146	10010010	01101101	109	m
7.	‡	135	10000111	01111000	120	x
8.	ˆ	136	10001000	01110111	119	w
9.	†	134	10000110	01111001	121	y
10.	¯	175	10101111	01010000	80	P
11.	¾	190	10111110	01000001	65	A
12.	¾	190	10111110	01000001	65	A
13.	®	174	10101110	01010001	81	Q
14.	Ô	212	11010100	00101011	43	+
15.	™	153	10011001	01100110	102	f
16.	µ	181	10110101	01001010	74	J
17.	Æ	198	11000110	00111001	57	9
18.	´	180	10110100	01001011	75	K
19.	»	187	10111011	01000100	68	D
20.	“	147	10010011	01101100	108	1
21.	¯	175	10101111	01010000	80	P
22.	Â	194	11000010	00111101	61	=
23.	Â	194	11000010	00111101	61	=
24.	ˆ	136	10001000	01110111	119	w

At the end of Process 1, an intermediate plain text 1 is 5HYLumxwyPAAQ+fJ9KD1P==w

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

TABLE 6

PROCESS 2 USING CROSS CHANGE & TRANSFORMATION

Sl. No	Intermediate Plain Text 1	Cross Change by even position	Cross Change by odd position	Transformation 1	Transformation 2
1.	5	H	5	5	Q
2.	H	L	Y	L	J
3.	Y	m	u	Y	f
4.	L	w	x	H	+
5.	u	P	y	u	9
6.	m	A	A	w	1
7.	x	+	Q	x	D
8.	w	J	f	m	k
9.	y	K	9	y	P
10.	P	1	D	A	w
11.	A	=	P	A	=
12.	A	w	=	P	=
13.	Q				
14.	+				
15.	f				
16.	J				
17.	9				
18.	K				
19.	D				
20.	1				
21.	P				
22.	=				
23.	=				
24.	w				

At the end of Process 2, an intermediate plain text 2 is 5LYHuwxmyAAPQJf+91DKPw==

TABLE 7

PROCESS 3 USING RIJNDAEL ALGORITHM

Sl. No	Intermediate Plain Text 2	ASCII Value	Key	ASCII Value	Original Message
1.	5	53	8	56	T
2.	L	76	3	51	r
3.	Y	89	6	54	a
4.	H	72	1	49	n
5.	u	117	4	52	s
6.	w	119	9	57	m
7.	x	120	5	53	i
8.	m	109	7	55	t
9.	y	121			
10.	A	65			
11.	A	65			
12.	P	80			
13.	Q	81			
14.	J	74			
15.	f	102			
16.	+	43			
17.	9	57			
18.	1	108			
19.	D	68			
20.	K	75			
21.	P	80			
22.	w	119			
23.	=	61			
24.	=	61			

At the end of Process 3, an original message “Transmit” is retrieved.

F. Advantages

- Capable of maximum (256) ASCII values for both input message and key.
- Provide multi-level encryption for plaintext.
- Lengthy unknown cipher text gives efficient protection to the encrypted message.
- Easy to compute as well as hard to analyse.
- Brute force attack needs 256! attempts to recover the plaintext.
- Same key should be used for consuming time.

IV. CONCLUSION

The major importance of this paper is given to the privacy issue of information, when the data should be transmit between sender and receiver securely. To achieve this privacy, a sort of efficient techniques to be taken while in encrypting the data. In this paper, a well defined multi level encryption algorithm is handled. A random key value is taken for every input character. The proposed work converts each plaintext character and random key value into its ASCII value in first step. For replacement of characters, a well known Rijndael algorithm method is invoked. After that process, genetic programming technique of cross change is involved. Transformation is capable for transposing the position of every cross changed characters. Finally, encode the data bits through the basic arithmetic and logic operations. In this method, shift and complement operations are performed, it covers all needed aspects of cryptographic security issues. The proposed is capable of maximum ASCII values for both plaintext and private key values.

REFERENCES

[1] William Stallings, 2007. Cryptography and Network Security. 3rd Edition. Pearson Education.
[2] Manoj Kumar, 2008. Cryptography & Network Security. 3rd Edition. Krishna Prakasan Media.
[3] Cryptography [Online]. Available: http://crypto.interactive – maths.com/codes-and-ciphers.html.
[4] R. Venkateswaram and V. Sundaram, "Information security: text encryption and decryption with poly substitution method and combining features of cryptography", International Journal of Computer Applications, vol.3, no.7, pp.28-31.
[5] Swati Mishra and Siddarh Bali, "Public key cryptography using genetic programming", International Journal of Recent Technology and Engineering, vol.2, no.2, pp.154-159.
[6] S.G. Srikantaswamy and H.D. Phaneendra, "A cipher design using the combined effect of arithmetic and logic operations with substitutions and transposition techniques", International Journal of Computer Applications (0975-8887), Sep.2011, vol.29, no.8, pp.34-36.
[7] S.G. Srikantaswamy and H.D. Phaneendra, "Enhanced onetime pad cipher with more arithmetic and logical operations with flexible key generation algorithm", International Journal of Network Security & Its Applications (IJNSA), Nov.2011, vol.3, no.6, pp. 243-248.
[8] S.G. Srikantaswamy and H.D. Phaneendra, "Improved Caesar cipher with random number generation technique and multistage encryption", International Journal on Cryptography and Information Security (IJCIS), Dec.2012, vol.2, no.4, pp.39-49.
[9] B. Bazith Mohammed, "Automatic Key Generation of Caesar Cipher", International Journal of Engineering Trends and Technology (IJETT), Dec. 2013, vol. 6, no. 6, pp.337-339.
[10] Govind Prasad Arya, Aayushi Nautiyal, Ashish Pant, Shiv Singh and Tishi Handa, "A cipher design with automatic key generation using the combination of substitution and transposition techniques and basic arithmetic and logic operations", The SIJ Transactions on Computer Science Engineering & its Applications (CSEA), Mar. 2013, vol. 1, no. 1, pp. 21-24.
[11] Devendra Prasad, Govind Prasad Arya, Chirag Chaudhary, Vipin Kumar, "Encipher A text encryption and decryption technique using substitution-transposition and basic arithmetic and logic operation", International Journal of Computer Science and Information Technologies (IJCSIT), 2014, vol. 5 (2).